Infraestrutura de Software 2023.2 - Final e 2a Chamada

1. O sistema operacional PintOS adota a política de escalonamento *round-robin*. Como funciona esta política? (2,0)
2. A política de escalonamento *Multiple Feedback Queue (*MFQ - Múltiplas Filas com Retroalimentação) é uma técnica melhor que *round-robin* para sistemas interativos? Explique. (2,0)
3. Faça um resumo das etapas que ocorre (hardware e sistema operacional) quando acontece uma falta de página (*page fault*) (2,0)
4. Cliente-Servidor e *Peer-to-Peer* são arquiteturas representativas para sistemas distribuídos. Explique-as (2,0).
5. Um sistema de som digital é composto por 3 *threads:* canal1, canal2, e canal3. Assuma que cada *thread é* implementada em um laço infinito. Durante uma iteração (repetição), cada *thread* deverá enviar seus dados para a função controladorSom(int dados[]) que representa o controlador de som. Tal controlador precisa ser usado somente por uma *thread* por vez, ou seja, não é permitido usar simultaneamente a função por mais de um canal. Considere um código qualquer dentro do laço de cada thread, mas precisa chamar a função controladorSom. **Implemente as threads canal1, canal2 e canal3 usando pthreads. Além da implementação, explique brevemente se exclusão mútua é uma técnica apropriada para esse problema.** (2,0)  
   Obs: Não precisa mostrar a criação das threads, mas você deve demonstrar a utilização de outros recursos necessários para a resposta.

int pthread\_mutex\_unlock(pthread\_mutex\_t \*mutex); int pthread\_mutex\_lock(pthread\_mutex\_t \*mutex); int pthread\_cond\_wait(pthread\_cond\_t \*cond, pthread\_mutex\_t \*mutex); int pthread\_cond\_signal(pthread\_cond\_t \*cond);int pthread\_cond\_broadcast(pthread\_cond\_t \**cond*); int pthread\_mutex\_init(pthread\_mutex\_t \*mutex, const pthread\_mutexattr\_t \*attr);int pthread\_mutex\_destroy(pthread\_mutex\_t \*mutex); int pthread\_join(pthread\_t *thread*, void \*\**value\_ptr*); int pthread\_barrier\_wait(pthread\_barrier\_t \*barrier); int pthread\_barrier\_init(pthread\_barrier\_t \* barrier, const pthread\_barrierattr\_t \* attr, unsigned int count );